## 

**Hunter Schafer** 



## ETL Assessment

#### The Good

- Lecture + Jupyter Notebooks
- Learn a lot on the HW
- Guided practice in section
- Identifying necessary material from the lectures
- Course staff is helpful and approachable

#### To Improve

- Project release date and clearer expectations
  - Part 0 Feedback
- Get HW grades faster
- Clearer feedback on HW
- Clearer expectations in section

### Exam 1 Overview

- 1. Programming in Python w/ Files
- 2. Tabular Data
- 3. Data Visualization + Machine Learning
- 4. Classes and Objects
- 5. Efficiency

For each type, we will explain the general idea and resources you might want to look at

 The list of resources is not a limitation of what will be on the problem, but just there to help you study

## 1 - PythonProgramming

#### General Idea

Write a function that takes a file name and does something while reading the file

- Lectures 1-4
- Section 1
- HW1

## 2- Tabular Data

#### General Idea

Write a function that uses tabular data

- A. In the list of dictionaries format
- B. In the pandas DataFrame format

- Lectures 4-7
- Section 2
- HW2 + HW3

## 3 - Data Visualization and Machine Learning

#### General Idea

#### Data Visualization

Write seaborn code to draw a graph and draw by hand

#### Machine Learning

Interpret sklearn code and explain ML concepts

- Lectures 8-9
- Section 3
- HW3

## 4 - Classes and Objects

#### General Idea

- Write a class to represent some data
- Write client code that uses an instance of that class

- Lecture 10
- Lectures 11-12 can be helpful, but a bit too focused on NLP
- Section 4
- HW4 (focus on classes and objects, less TF-IDF and search)

## 5 - Efficiency

#### General Idea

Look at code and write down its run-time in Big-O efficiency

Must write "tightest" bound

#### Resources

Lecture 13 mostly, but Lecture 14 also has some examples

### Tightness

Consider the following code example, what is it's run-time?

```
def fun(nums):
    total = 0
    for n in nums:
        total += n
    for n in nums:
        total += 2 * n
    return total
```

The formal definition of Big-O allows for answers like  $O(n^{12})$ , this is not what we are interested in. We want you to write down the "smallest" run-time that fits. In this case O(n) would be the answer.

## Machine Learning

#### Remember the standard ML setup

```
X = iris[iris.columns != 'species']
y = iris['species']
X_train , X_test , y_train , y_test = \
    train_test_split(X, y, test_size=0.2)
model = DecisionTreeClassifier()
model.fit(X_train , y_train)
y_test_predict = model.predict(X_test)
print(accuracy_score(y_test, y_test_predict))
```



Think &

1.5 minutes

Why do we need a test set when doing machine learning? Explain your answer in one or two sentences.

Everyone will enter in a free response answer, I will select 4 or so at random and then in the Pair stage, you try to identify which answer you think is best.

The answers I select will be anonymous.

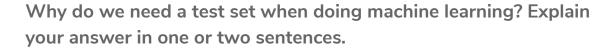






Pair 22

2 minutes



For the answers shown on the device, choose which one you think is best. Talk with your group about why you think that answer is best. Some things to consider when evaluating an answer

- Does the response answer the question?
- Is there anything missing?
- What can be removed from the response.





### ML Recap

For machine learning, we don't want you to memorize the exact lines of code and be able to write them. What we do want is you to have an understanding of WHY we need each step in the process.

This is important to us because the ML library you use will change every couple of years, but the ideas we have talked about will be around for a long while.

## Brain Break



#### Sections

Practice material covered in 1 in a context where a TA can help you.

The emphasis is still on you learning by doing.

#### **Homeworks + Project**

With the scaffolding from 1 and 2, you are probably now capable to tackle the homeworks. These will be complex and challenging, but you'll continue to learn by doing.

#### Lectures

Introduced to material for the first time. Mixed with activities and demos to give you a chance to **learn by doing.** 

No where near mastery yet!

# Classes and Objects

- The theme here is you learn by doing
- We want to give you some time to practice, and then we will go over how to solve as a class
- The problem we are handing out looks like it will on the exam on Friday
- Time to work
  - 3 minutes by yourself
  - 6 minutes working together with a group
    - Probably won't be done in 3 minutes, explain your process to your group



